



# Virtual Sculpture with Haptic Feedback

Renaud Blanch

## ► To cite this version:

Renaud Blanch. Virtual Sculpture with Haptic Feedback. Synthèse d'image et réalité virtuelle [cs.GR]. 2000. inria-00598408

**HAL Id: inria-00598408**

**<https://inria.hal.science/inria-00598408>**

Submitted on 6 Jun 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Rapport de stage effectué au laboratoire iMAGIS<sup>1</sup>

Renaud BLANCH  
juillet - septembre 2000

---

## Encore merci

Je tiens ici à remercier encore une fois toute l'équipe d'iMAGIS que j'ai côtoyée cet été, je garderai un agréable souvenir de mon passage dans leurs murs.

Merci surtout à Eric pour ses conseils, les discussions que nous avons pu avoir tout au long de ce stage et la grande liberté qu'il m'a laissée; merci à ses responsables de thèse, Marie-Paule Cani et Jean-Dominique Gascuel, qui ont pris le temps de suivre mon travail. Merci enfin à tous ceux qui, par leurs remarques constructives et leurs suggestions pertinentes, m'ont guidé dans ce travail.

---

## Quelques mots ...

*La forme du présent rapport pourra surprendre; il est en fait destiné à deux lectures. Celle des gens avec qui j'ai travaillé cet été à qui elle apportera un compte-rendu suffisamment détaillé, je l'espère, de mon travail ainsi qu'une bibliographie des domaines utiles.*

*Par ailleurs, il est aussi destiné à Supélec pour rendre compte de mon activité durant le stage et contient à ce titre une présentation plus large du laboratoire et du contexte dans lequel j'ai été amené à choisir ce stage et les impressions et enseignements que j'en ai tiré.*

## Pourquoi ?

Comptant effectuer dans le cadre d'aménagements de l'enseignement entre Supélec (où j'ai terminé en 1999-2000 ma seconde année) et la faculté de sciences d'Orsay un DEA d'informatique en parallèle de ma troisième année, j'ai voulu profiter de mon stage d'été pour découvrir un laboratoire de recherche en informatique.

Par ailleurs, l'image et le graphisme font partie de mes centres d'intérêt et l'opportunité qui s'est présentée au

sein d'iMAGIS m'a permis de concilier au mieux ceux-ci et les objectifs du stage.

## Quoi ?

Le cadre de mon stage était l'ajout de fonctionnalités liées à l'interface avec l'utilisateur à un logiciel de sculpture virtuelle<sup>2</sup>. Ce logiciel est développé dans le cadre de sa thèse par Eric Ferley, doctorant en troisième année, à iMAGIS.

J'ai travaillé sur deux aspects qui seront présentés ici successivement; tout d'abord la mise au point d'une technique permettant les couper/copier/coller de volume au sein de l'application.

Par ailleurs, j'ai interfacé l'application avec un dispositif de pointage 3D à retour d'effort. Cette mise en place a soulevé des problèmes pour lesquels quelques solutions sont proposées.



FIG. 1 – Exemple de sculpture réalisée grâce au logiciel par Marie-Paule Cani

---

<sup>1</sup>iMAGIS-GRAVIR/IMAG est un projet commun CNRS, INRIA, UJF, INPG. serveur : [www-imagis.imag.fr/](http://www-imagis.imag.fr/)

---

<sup>2</sup>une présentation de l'application est disponible à l'adresse : [www-imagis.imag.fr/Membres/Eric.Ferley/sculpture/sculpt.html](http://www-imagis.imag.fr/Membres/Eric.Ferley/sculpture/sculpt.html)

# 1 Sélection

Tout d'abord sont donnés ici les éléments sur l'application de sculpture virtuelle qui sont nécessaires à la compréhension du cadre dans lequel s'inscrit mon travail.

Ensuite vient la description de la solution mise en place pour gérer les couper/copier/coller au sein de l'application.

## 1.1 Le logiciel

Dans cette partie, le logiciel est présenté rapidement. Pour une description plus détaillée, on pourra se reporter à [5, 6] disponibles à l'adresse :

[www-imagis.imag.fr/Publications/index.html](http://www-imagis.imag.fr/Publications/index.html)  
et à leurs bibliographies.

### Une surface implicite

La surface sur laquelle travaille l'utilisateur est une isopotentielle d'un champs scalaire échantillonné dans l'espace. La valeur de ce champs est donc stockée aux coins des cubes où il est défini et interpolée de manière trilineaire dans les cubes. Cette valeur indique pour chaque point, par comparaison avec l'isovaleur choisie, s'il est à l'intérieur ou à l'extérieur de la surface.

L'isosurface est par la suite triangulée pour pouvoir être affichée grâce à un algorithme classique de Marching-Cubes.

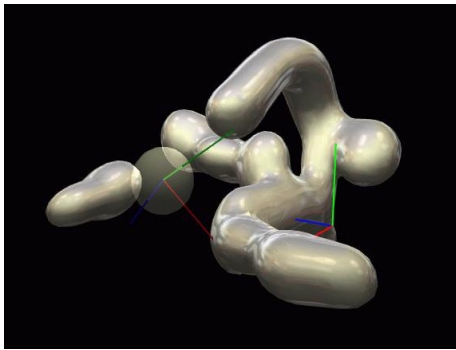


FIG. 2 – Capture d'écran de l'application

### Sculpture

En pratique, la sculpture s'effectue en modifiant localement le potentiel. Les outils utilisés sont, eux aussi, définis par leur potentiel et par la manière dont ils le combinent avec celui qui définit la surface.

Typiquement, ils ajoutent ou soustraient leur contribution et permettent ainsi de modifier le potentiel en

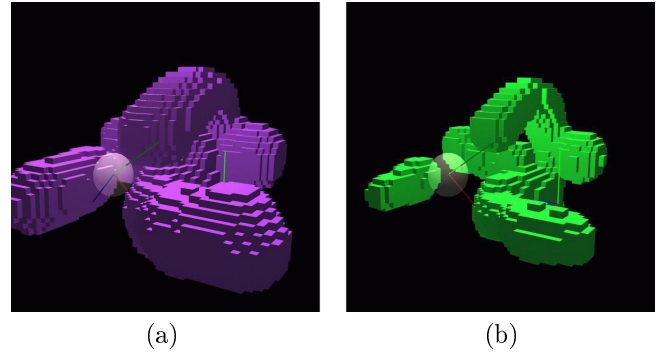


FIG. 3 – Grille cubique : (a) Cubes dans lequel le potentiel est défini pour l'objet de la figure 2 - (b) Cubes en intersection avec l'isosurface pour le même objet.

déposant ou retirant de "la matière". Des compositions plus subtiles existent qui permettent des déformations simulant le fait de repousser la matière sur le passage de l'outil.

La possibilité d'utiliser par la suite un objet modélisé au sein de l'application comme nouvel outil (appelé *outil discret*) élargit très rapidement les possibilités de création. Son potentiel est alors rééchantillonné lors de l'application pour calculer en fonction de son échelle et de son orientation sa contribution en potentiel sur la grille de la sculpture.

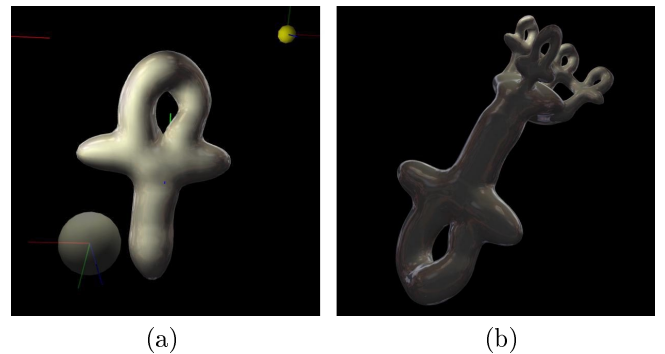


FIG. 4 – Outil discret : (a) Volume sculpté grâce à l'outil sphérique de base - (b) Sculpture réalisée en utilisant ce nouveau volume comme outil.

### Implémentation

Le logiciel est développé en C++ et utilise la librairie OpenGL ainsi que la STL de SGI. La machine utilisée pour son évaluation est une Onyx2 de SiliconGraphics comportant deux processeurs R10000 à 195 MHz et une carte graphique InfiniteReality2 mais il tourne aussi sur PC sous WindowsNT/Linux.

## 1.2 Edition

La sélection d'une partie de la surface d'une sculpture pour pouvoir la couper ou la copier en vue de sa réutilisation n'est pas simple. En effet, il faudrait retrouver, à partir de la surface, le potentiel qui permet de la générer puisqu'elle est définie implicitement.

L'idée qui est venue est celle d'une sélection volumique qui permet de calculer simplement un potentiel adéquat pour la reconstruction de la surface englobée dans ce volume.

### Calque de sélection

Pour sélectionner un volume, on travaille sur un nouvel objet similaire à la sculpture elle-même affiché en transparence. On utilise donc les outils standards pour le créer en déposant du potentiel qui n'a plus le sens de "densité de matière" comme lors de la sculpture mais plutôt de "degré de sélection" de la zone qu'il recouvre.

Cet objet, par analogie avec les logiciels classiques de retouche d'image, a été nommé *calque de sélection*. La possibilité d'utiliser les outils habituels pour créer et modifier ce calque lui confère une grande souplesse d'utilisation. Il est par ailleurs échantillonné sur la même grille spatiale que la sculpture.

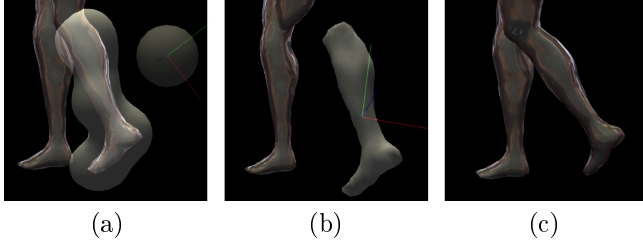


FIG. 5 – Calque de sélection : (a) La sculpture et le calque de sélection englobant une jambe - (b) La jambe coupée et la sélection affectée comme outil - (c) La jambe collée avec un angle différent

### Calcul de la sélection

Le calcul de la sélection s'effectue simplement, il s'apparente au calcul de l'intersection des volumes définis par la sculpture et par le calque de sélection. Les surfaces étant définies par leurs potentiels, cette *intersection* consiste à créer un nouvel objet dont le potentiel est défini aux points d'échantillonnage par :

$$V_{sel} = \min(V_{sculpt}, V_{layer})$$

où :

- $V_{sel}$  est le potentiel de la sélection calculé,
- $V_{sculpt}$  est le potentiel qui définit la sculpture,
- $V_{layer}$  définit le calque de sélection.

En pratique, on parcourt le volume du calque et on crée au fur et à mesure de ce parcours une nouvelle structure de données identique aux deux autres comportant comme valeur de potentiel le minimum des deux valeurs.

Le potentiel ainsi créé garde donc de bonnes propriétés de continuité et définit bien une surface conforme à la partie de sculpture sélectionnée.

### Copier, couper, coller

Ces opérations de bases sont implémentées simplement : on a vu qu'on avait la possibilité d'utiliser comme outil un objet lui même.

*Copier* consiste donc à calculer l'intersection et à utiliser celle-ci comme nouvel outil après l'avoir centrée autour du point d'application de l'outil. On peut alors l'utiliser en changeant son échelle et son orientation avant de l'appliquer pour effectuer un *coller*.

Pour *couper*, on utilise exactement le même principe en soustrayant la sélection de la sculpture avant de la réutiliser comme outil pour par exemple la déplacer (cf. figure 5).

### Remplir

Une dernière fonctionnalité a été ajoutée : la possibilité de remplir la sélection, c'est à dire de calculer l'*union* de la sculpture et du calque de sélection. Elle s'obtient en prenant pour nouveau potentiel qui définit la sculpture :

$$\max(V_{sculpt}, V_{layer})$$

On évite ainsi des problèmes complexes de fonction de mélange des deux potentiels (cf. figure 6).

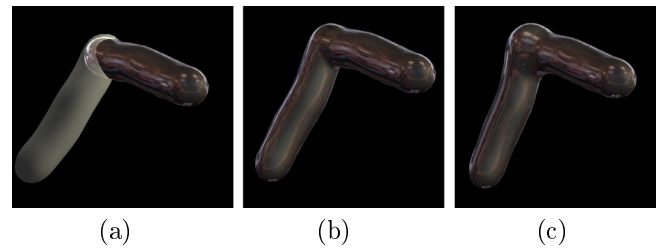


FIG. 6 – Utilisation du remplissage : (a) L'objet et le calque de sélection - (b) Après remplissage : la surface englobe exactement l'union des volumes - (c) Résultat obtenu en sommant les potentiels : une bosse se forme au raccord.

Cette fonctionnalité peut aussi être utile par exemple lorsque l'on veut créer deux objets proches sans que leurs surfaces finissent par s'interconnecter lorsque leurs potentiels se somment.

## 2 Rendu haptique

Le but du travail présenté dans cette partie est d'interfacer le programme de sculpture virtuelle avec un dispositif à retour d'effort. Il s'agit donc de réaliser, selon l'expression consacrée, un *rendu haptique* d'une surface ; celle-ci étant ici définie implicitement comme isopotentielle d'un champ scalaire échantonné sur une grille cubique.

Plusieurs approches ont été tentées pour le rendu haptique, utilisant l'API GHOST fournie par Sensable à différents niveaux et donnant des résultats divers. Elles sont exposées ici.

Par ailleurs, des problèmes de stabilité apparaissent entraînant des vibrations indésirables dans la simulation. Nous avons donc mis en œuvre un filtrage sur les informations issues du PHANTOM pour les atténuer. Ce traitement est détaillé à la fin de cette partie.

### 2.1 Modèle polygonal local dynamique

L'idée qui sous-tend cette approche est d'utiliser les objets préexistants dans l'API GHOST qui permet de communiquer avec le périphérique en les adaptant à notre usage. La méthode présentée ici est très similaire à celle présentée dans [8].

L'isosurface étant triangulée pour les besoins de l'affichage et les collections de triangles étant des objets définis par GHOST, nous avons utilisé les triangles de la surface pour que l'API se charge de calculer une force résultante lors du contact de ceux-ci.

Le calcul de cette force utilise un modèle masse-ressort paramétrable et une composante tangentielle introduite pour la friction [9].

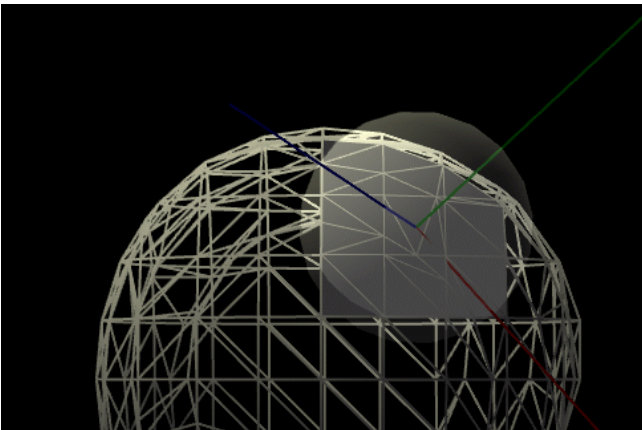


FIG. 7 – Exemple de triangles sélectionnés pour le rendu haptique. L'objet est en fil de fer et l'outil en transparence.

### Mise en œuvre

La contrainte forte sur la fréquence de mise à jour de la force (1kHz) limite pour la machine sur laquelle est effectuée l'implémentation le nombre de triangles utilisables à environ 200.

Les tests de collisions sont en effet optimisés par GHOST dans le cas de modèles polygonaux statiques grâce à une partition spatiale précalculée. Cependant, dans le cas qui nous intéresse, la surface évoluant au cours de l'exécution du programme, l'usage de partition spatiale n'est pas possible. Seuls sont donc envoyés au PHANTOM les triangles situés au voisinage du pointeur. Ils sont déterminés de la manière suivante :

- L'examen des 27 cubes de la grille environnants le pointeur permet de collecter dans une liste `newcubes` les cubes en intersection avec l'isosurface.
- Un parcours de la liste des cubes affichés `cubes` à l'étape précédente permet de :
  - supprimer dans celle-ci les cubes qui n'ont plus lieu d'être, libérant ainsi de la place pour l'insertion des nouveaux cubes et la réutilisation des triangles,
  - supprimer parmi les nouveaux cubes potentiels ceux qui sont déjà présents.
- Enfin, les nouveaux cubes effectifs sont insérés dans la liste `cubes` et les coordonnées des triangles correspondants sont envoyés pour le rendu haptique.

### Implémentation

On utilise pour implémenter le modèle local les classes de GHOST `gstTriPolyMesh` et `gstTriPolyMeshHaptic`. Elles sont initialisées avec une collection de triangles dont les coordonnées sont modifiées indépendamment en fonction des besoins déterminés par le parcours ci-dessus.

### Rendu

La sélection des triangles fonctionne bien en temps interactif puisque la fréquence de mise à jour de ceux-ci est au minimum de 300 Hz. Le pointeur ne risque donc pas de traverser la surface avant que les triangles n'aient été "mis en place" à son voisinage.

Cependant un problème qui semble lié à l'utilisation de GHOST<sup>3</sup> dans un domaine pour lequel il n'est pas conçu rend cette approche peu convaincante. Le stylet semble en effet "coller" aux triangles, une force assez importante étant nécessaire pour le séparer des triangles lorsqu'il est entré en contact avec eux.

<sup>3</sup>[8] fait lui aussi état de difficultés rencontrées lors de l'utilisation du tool-kit de Sensable

## 2.2 Force dérivée du modèle volumique

Nous avons donc finalement décidé de nous affranchir des polygones introduits somme toute artificiellement pour la représentation graphique. L'API offre la possibilité d'envoyer directement au bras une force calculée. C'est cette solution qui a été retenue, l'application ayant la charge de calculer et de fournir une force cohérente.

L'information "volumique" est suffisante pour calculer cette force puisque par exemple, une simple lecture de la valeur du potentiel à un endroit permet de se situer entre l'intérieur et l'extérieur de la surface. Avila et Sobierajski ont donné des pistes pour le rendu ponctuel d'information volumique et d'isosurfaces [2, 1]. La méthode ponctuelle utilisée ici en diffère par les fonctions de transfert entre potentiel et force.

Deux types de forces ont été implémentées : l'une est uniquement fonction de la position du centre de l'outil (et de sa vitesse pour une composante de friction) ; l'autre tient compte de la géométrie de l'outil en intégrant l'information dans le volume de l'outil.

### Viscosité

Le terme de viscosité opposé au déplacement est simplement :

$$\vec{f}_v = -\alpha V(\vec{p}) \vec{v}$$

où :

- $\alpha$  est une constante positive,
- $V(\vec{p})$  est le potentiel scalaire interpolé au point  $\vec{p}$  position du stylet,
- $\vec{v}$  est la vitesse du stylet.

Le fait de faire intervenir le potentiel pour pondérer la force permet de ressentir la densité de la matière par la résistance qu'elle oppose au déplacement. Cependant l'information de surface n'est pas présente dans la force calculée ici. Elle est donc combinée avec l'une de celles présentées ci-dessous.

### Force ponctuelle

Pour rendre compte d'une interaction avec la surface, la force doit être normale à celle-ci. Elle sera donc dirigée selon le gradient du potentiel. Sa norme doit par ailleurs présenter un front montant assez raide au voisinage de l'isosurface, l'information de distance à la surface étant estimée par l'écart entre le potentiel et l'isovaleur comme suggéré par [2]. Enfin, la norme est bornée pour des raisons techniques. Nous aboutissons alors à la force suivante :

$$\vec{f}_p = -\frac{\text{grad}(\vec{V})}{\|\text{grad}(\vec{V})\|} \min\left(\beta \left(\frac{V}{iso}\right)^e, maxval\right)$$

où :

- $\text{grad}(\vec{V})$  est le gradient du potentiel interpolé au point courant,
- $\beta$  est une constante positive,
- $V$  est le potentiel interpolé au point concerné,
- $iso$  est la valeur définissant l'isosurface qui permet de normaliser le potentiel sur celle-ci,
- $e$  est un exposant réel positif qui permet de régler la pente de la norme au voisinage de l'isosurface,
- $maxval$  est la borne de la norme de la force envoyée au bras.

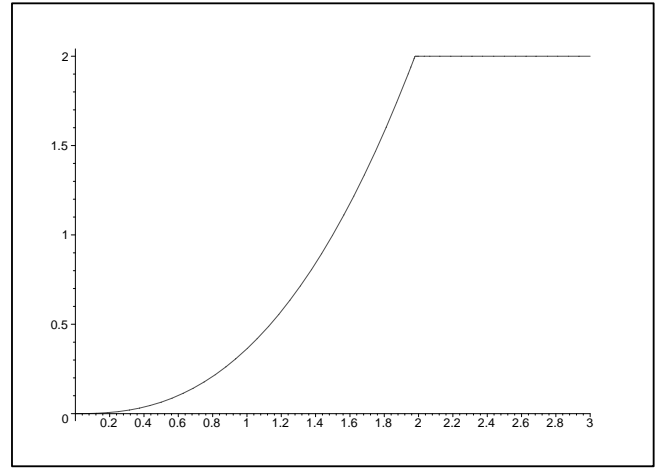


FIG. 8 – Norme de la force en fonction du potentiel.

Cette force se calcule pratiquement à la fréquence de 1000 Hz requise pour une bonne simulation haptique. Le toucher obtenu permet une localisation précise du centre de l'outil par rapport à l'isosurface.

### Force intégrée

La force calculée ici est la moyenne, effectuée sur les cubes du volume d'application de l'outil, des forces portées par le gradient du potentiel. Leur norme est proportionnelle au potentiel de l'outil et au potentiel de l'objet. On obtient donc une force de répulsion dès que l'objet et la zone d'influence de l'outil sont en intersection.

Cette force, malgré de bonnes propriétés au niveau de la perception présente des inconvénients. Elle est tout d'abord lente à calculer notamment lorsque l'outil devient grand. Il faudrait alors sans doute extrapoler la force entre deux calculs successifs ce qui ne semble pas évident a priori. Par ailleurs, si l'outil s'étend sur des zones où le gradient du potentiel varie fortement en direction, les forces calculées successivement peuvent

présenter de fortes discontinuités et conduire à des comportements instables.

## Rendu

La force utilisée pour le rendu est une somme pondérée de la force de frottement et de l'une des deux autres. Les coefficients de la combinaison varient suivant que l'outil est appliqué ou non. Quand il est appliqué, c'est la force de frottement qui est prédominante ce qui permet de pénétrer dans la surface tout en ayant une bonne sensation de la densité de la matière à l'endroit où l'on se trouve. Quand l'outil n'est pas appliqué, c'est la force de surface qui est prépondérante, permettant ainsi de se situer grâce au contact de l'objet.

La transition entre les deux combinaisons est par ailleurs réalisée progressivement pour éviter les discontinuités de la force.

Guidé par le retour d'effort, et grâce au fait que le dispositif de pointage est vraiment en 3 dimensions, l'utilisation du logiciel est simplifiée et surtout beaucoup plus intuitive.



FIG. 9 – Exemple de modèle que j'ai réalisé en quelques heures à l'aide du logiciel et du PHANToM.

## 2.3 Traitement des vibrations

La simulation retenue repose donc sur le calcul, effectué à 1000 Hz environ, d'une force ponctuelle liée à la position et à la vitesse du pointeur. Cependant, il apparaît des vibrations indésirables dues à la force qu'oppose l'utilisateur au stylet. Ce problème a été formalisé et résolu par ailleurs [4, 7] dans le cas simple du *mur virtuel* présenté brièvement ci-dessous.

Pour le cas qui nous intéresse, le problème est plus complexe étant donné la forme a priori quelconque que peuvent prendre le potentiel et son gradient. Nous avons donc développé une solution basée sur une approche qualitative et intuitive du comportement désiré.

La méthode de filtrage mise au point pour supprimer ces vibrations est présentée ci-après.

### Origine des vibrations

Les vibrations observées ont des origines multiples déjà étudiées dans la littérature classique. L'une d'entre elle s'exprime simplement : la force calculée à un point fixe donné est orientée suivant la descente du gradient du potentiel. Elle a donc tendance à repousser l'outil vers une région où le potentiel a des valeurs inférieures, où la force a une norme plus faible. L'utilisateur a lui tendance à s'opposer au déplacement de l'outil.

En simplifiant, utilisons un modèle de mur constitué par un demi-espace (l'extérieur du mur) dans lequel la force est nulle et l'autre demi-espace (le mur) dans lequel la force est constante et dirigée vers l'extérieur. Les vibrations apparaissent à la frontière : lorsque l'outil pénètre dans le mur, la force le repousse à l'extérieur et la résistance appliquée par l'utilisateur n'a plus de compensation. L'outil rentre alors à nouveau dans le mur et le régime oscillant se poursuit.

C'est ce type de phénomène qui apparaît et rend la simulation peu utilisable.

### Solution mise en place

L'idée qui vient alors est de filtrer la force en utilisant un filtre passe-bas pour couper ces variations en hautes fréquences. Cependant, alors que la force envoyée au bras dépend directement de la position, la position est, elle, fonction de la force et de la réaction l'utilisateur. En filtrant simplement la force, on ne tient pas compte de l'action de celui-ci.

La donnée qui intègre l'action conjointe de l'utilisateur et de la force envoyée au bras est en fait simplement la position de celui-ci. C'est donc la position que nous avons choisi de filtrer.

Le fait de n'utiliser l'information issue du PHANToM qu'à travers un filtre est déjà présent dans [3] sous le

nom de couplage virtuel mais le filtrage mis en place ici est particulier : il n'est pas linéaire.

### Fausse position

La position utilisée pour le calcul de la force est donc une fausse position qui est issue du filtrage de la position réelle renvoyée par le bras. Cette position est amortie exponentiellement avec une constante de temps choisie pour couper les hautes fréquences et ne peut donc pas osciller, la force calculée garde donc de bonnes propriétés de continuité.

Cependant, la mémoire sur la position introduite par l'amortissement exponentiel a des effets indésirables. Dans des gammes de fréquences plus basses (liées à la constante de temps choisie), des régimes oscillants de plus grande amplitude sont obtenus. Le filtrage est donc raffiné pour prendre effet uniquement dans les mouvements de faible amplitude. En pratique, le filtrage réalisé se décompose ainsi :

- Calcul de la position amortie exponentiellement :

$$\begin{aligned}\delta\vec{p} &= \vec{p}_r - \vec{p}_a(t-1) \\ \vec{p}_a(t) &\leftarrow \vec{p}_a(t-1) + \alpha \delta\vec{p}\end{aligned}$$

- Calcul de la fausse position comme une combinaison linéaire de la position réelle et de la position amortie, cette combinaison étant fonction de la distance qui les sépare :

$$\begin{aligned}\vec{p} &= \beta \vec{p}_a + (1 - \beta) \vec{p}_r \\ \text{avec } \beta &= \frac{\lambda}{\|\delta\vec{p}\| + \lambda}\end{aligned}$$

où :

- $\vec{p}_r$  est la position réelle renvoyée par le bras,
- $\vec{p}_a$  est la position amortie,
- $\vec{p}$  est la fausse position utilisée par le programme,
- $\alpha$  est la constante de temps de l'amortissement,
- $\lambda$  est une constante positive.

La combinaison est effectuée selon le principe suivant : si les positions sont proches, c'est que l'outil ne bouge pas ou bien qu'il vibre autour d'une position. C'est alors la position amortie qui est utilisée. Si par contre les positions sont éloignées, c'est qu'on effectue un déplacement volontaire où qu'on entre dans un mode oscillant de basse fréquence et de grande amplitude. C'est alors la position réelle qui est utilisée. La transition entre ces deux comportements est rendue continue en utilisant la distance entre les deux positions pour pondérer la combinaison.

### Propriété

Une propriété intéressante de cette fausse position est que la distance entre la position réelle (renvoyée par le PHANTOM) et la position utilisée pour le calcul de force et l'affichage est bornée. En effet :

$$\begin{aligned}\|\vec{p} - \vec{p}_r\| &= \beta \|\vec{p}_a - \vec{p}_r\| \\ &= \beta \|\delta\vec{p}\| \\ &= \frac{\lambda}{\|\delta\vec{p}\| + \lambda} \|\delta\vec{p}\| \\ &\leq \lambda\end{aligned}$$

En pratique, elle est rendue très faible et l'utilisateur ne perçoit pas le décalage existant. La fausse position proposée est donc intéressante puisqu'elle permet d'assurer une cohérence entre les représentations haptique et visuelle, nécessaire à l'immersion de l'utilisateur.

### Résultat

L'utilisation de la fausse position présentée, moyennant une bonne adaptation de la constante de temps de l'amortissement et des coefficients de la combinaison, supprime complètement les vibrations qui pouvaient être ressenties au niveau de la surface.

Par ailleurs, la force de viscosité étant fonction de la vitesse, elle a été filtrée elle aussi mais plus simplement. Un amortissement exponentiel de celle-ci suffit à supprimer les instabilités dues aux frottements.

### Références

- [1] R. Avila. Volume haptics. *Computer Graphics*, pages 103–123, July 1998. SIGGRAPH'98 Course Notes #01.
- [2] R. Avila and L. Sobierajski. A haptic interaction method for volume visualization. *Computer Graphics*, pages 197–204, Oct. 1996. Proceedings of Visualization'96.
- [3] J. E. Colgate, M. C. Stanley, and J. M. Brown. Issues in the haptic display of tool use, 1995. IROS'95.
- [4] R. E. Ellis, N. Sarkar, and M. A. Jenkinsy. Numerical methods for the force reflection of contact, 1996. ASME'96.
- [5] E. Ferley, M.-P. Cani, and J.-D. Gascuel. Practical volumetric sculpting, Sept. 1999. Proceedings of Implicit Surfaces'99 (Bordeaux, France).
- [6] E. Ferley, M.-P. Cani, and J.-D. Gascuel. Virtual sculpture (short paper), Sept. 1999. Proceedings of Eurographics '99 (Milano, Italy).
- [7] R. B. Gillespie and M. R. Cutkosky. Stable user-specific haptic rendering of the virtual wall, Nov. 1996. Proceedings of ASM IMECE.



- [8] O. Körne, M. Schill, C. Wagner, H. Bender, and R. Männer. Haptic volume rendering with an intermediate local representation, 1999. First International Workshop on the Haptic Devices in Medical Applications.
- [9] T. H. Massie and J. K. Salisbury. The phantom haptic interface : A device for probing virtual objects, 1994. Proceedings of ASME'94.